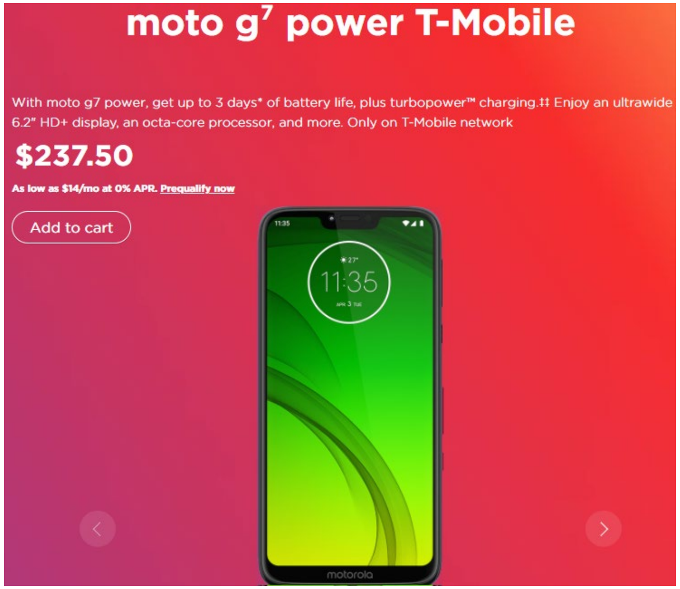
Exhibit G

Claim Chart – '818 Patent

US Patent 6,744,818 Versus Moto G7 Power Smartphones with H.264/AVC



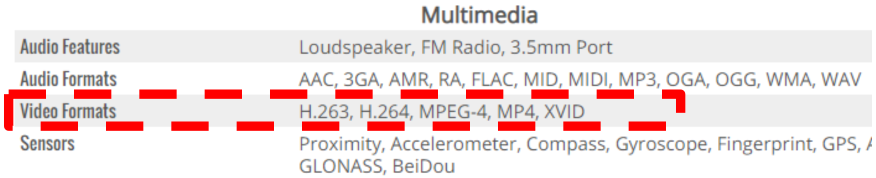
moto g7 power T-Mobile

With moto g7 power, get up to 3 days* of battery life, plus turbopower™ charging.‡‡ Enjoy an ultrawide 6.2" HD+ display, an octa-core processor, and more. Only on T-Mobile network

$237.50

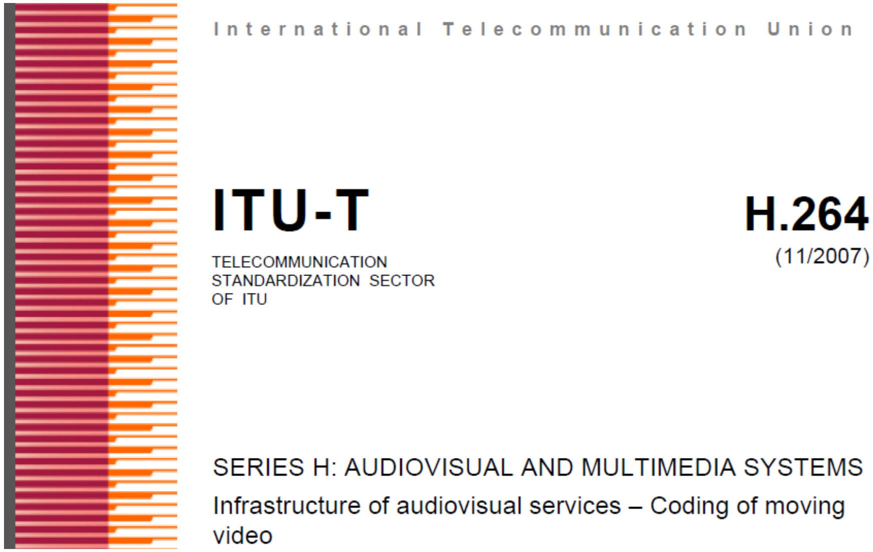
As low as $14/mo at 0% APR. Prequalify now

Add to cart



Multimedia

Audio Features	Loudspeaker, FM Radio, 3.5mm Port
Audio Formats	AAC, 3GA, AMR, RA, FLAC, MID, MIDI, MP3, OGA, OGG, WMA, WAV
Video Formats	H.263, H.264, MPEG-4, MP4, XVID
Sensors	Proximity, Accelerometer, Compass, Gyroscope, Fingerprint, GPS, GLONASS, BeiDou



International Telecommunication Union

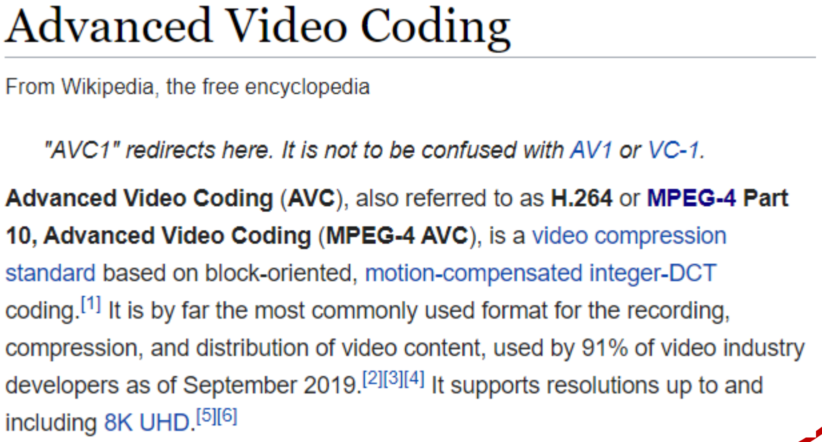
ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

H.264
(11/2007)

SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS

Infrastructure of audiovisual services – Coding of moving video



Advanced Video Coding

From Wikipedia, the free encyclopedia

"AVC1" redirects here. It is not to be confused with AV1 or VC-1.

Advanced Video Coding (**AVC**), also referred to as **H.264** or **MPEG-4 Part 10, Advanced Video Coding** (**MPEG-4 AVC**), is a video compression standard based on block-oriented, motion-compensated integer-DCT coding.[1] It is by far the most commonly used format for the recording, compression, and distribution of video content, used by 91% of video industry developers as of September 2019.[2][3][4] It supports resolutions up to and including 8K UHD.[5][6]

Preliminary Claim Chart Showing Infringement of Claim 1 of the U.S. Patent No. 6,744,818 by Lenovo

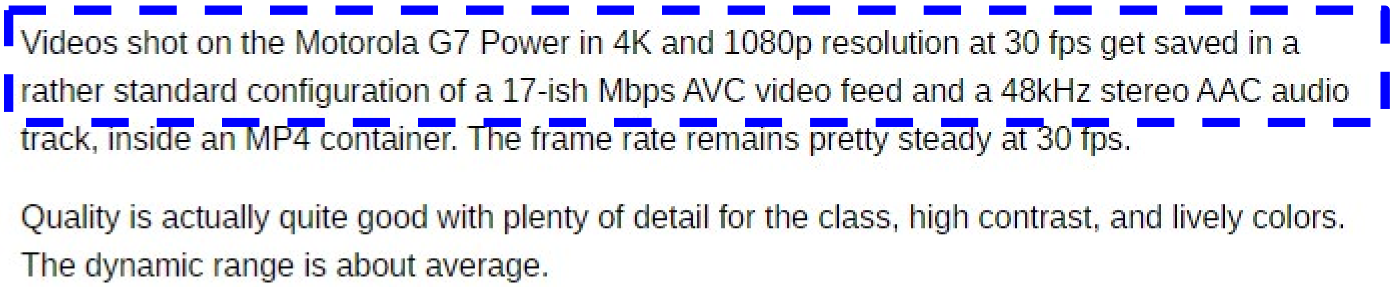
1. A video encoding system comprising:

a visual perception estimator adapted to estimate a perception threshold for a pixel of a current frame of a videostream;

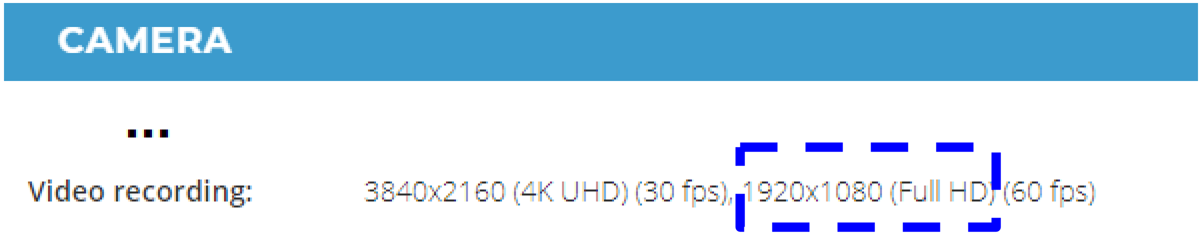
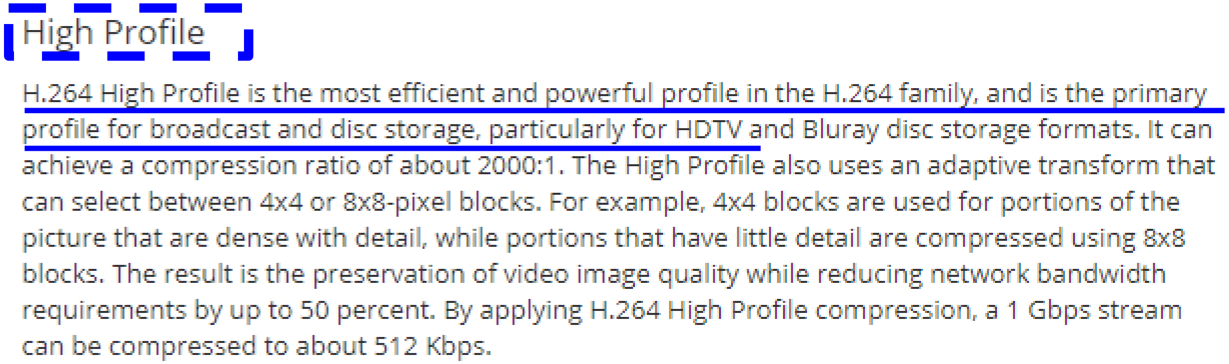
an encoder adapted to encode said current frame;

a compression dependent threshold estimator adapted to estimate a compression dependent threshold for said pixel at least from said perception threshold and information from said encoder; and

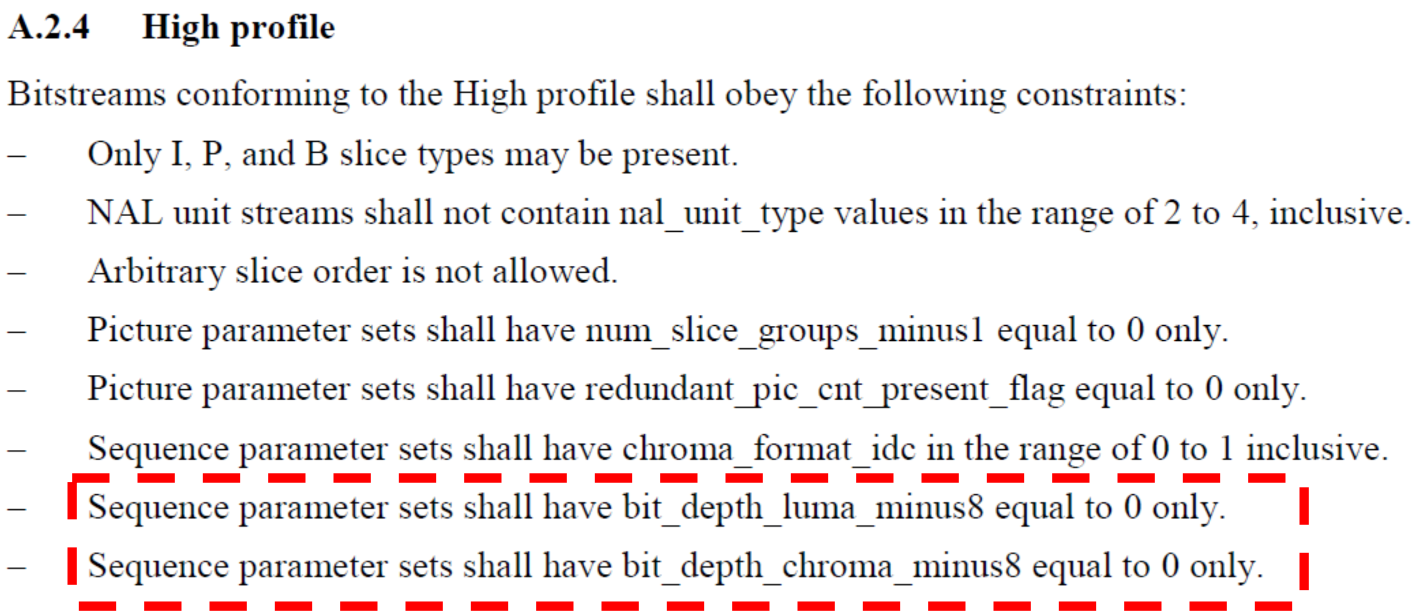
a filter unit adapted to filter said pixel at least according to said compression dependent threshold.

Claim 1	
A video encoding system comprising:	https://www.gsmarena.com/motorola_moto_g7_power-review-1889p5.php GSMArena team, 17 February 2019 Motorola Moto G7 Power review ...  Videos shot on the Motorola G7 Power in 4K and 1080p resolution at 30 fps get saved in a rather standard configuration of a 17-ish Mbps AVC video feed and a 48kHz stereo AAC audio track, inside an MP4 container. The frame rate remains pretty steady at 30 fps. Quality is actually quite good with plenty of detail for the class, high contrast, and lively colors. The dynamic range is about average.

Preliminary Claim Chart Showing Infringement of Claim 1 of the U.S. Patent No. 6,744,818 by Lenovo

Claim 1	
A video encoding system comprising:	G7 Power records with H.264/AVC and "High Profile": https://www.phonearena.com/phones/Motorola-Moto-G-Power_id11349  CAMERA ... Video recording: 3840x2160 (4K UHD) (30 fps), 1920x1080 (Full HD) (60 fps) "High Definition" uses the H.264 "High Profile": https://www.rgb.com/h264-profiles  High Profile H.264 High Profile is the most efficient and powerful profile in the H.264 family, and is the primary profile for broadcast and disc storage, particularly for HDTV and Bluray disc storage formats. It can achieve a compression ratio of about 2000:1. The High Profile also uses an adaptive transform that can select between 4x4 or 8x8-pixel blocks. For example, 4x4 blocks are used for portions of the picture that are dense with detail, while portions that have little detail are compressed using 8x8 blocks. The result is the preservation of video image quality while reducing network bandwidth requirements by up to 50 percent. By applying H.264 High Profile compression, a 1 Gbps stream can be compressed to about 512 Kbps.

Preliminary Claim Chart Showing Infringement of Claim 1 of the U.S. Patent No. 6,744,818 by Lenovo

Claim 1	
a visual perception estimator adapted to estimate a perception threshold for a pixel of a current frame of a videostream ;	https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-H.264-200711-S!!PDF-E&type=items pg. 286 The “High Profile” includes the use of bit_depth_luma and bit_depth_chroma:  **A.2.4 High profile** Bitstreams conforming to the High profile shall obey the following constraints: – Only I, P, and B slice types may be present. – NAL unit streams shall not contain nal_unit_type values in the range of 2 to 4, inclusive. – Arbitrary slice order is not allowed. – Picture parameter sets shall have num_slice_groups_minus1 equal to 0 only. – Picture parameter sets shall have redundant_pic_cnt_present_flag equal to 0 only. – Sequence parameter sets shall have chroma_format_idc in the range of 0 to 1 inclusive. – Sequence parameter sets shall have bit_depth_luma_minus8 equal to 0 only. – Sequence parameter sets shall have bit_depth_chroma_minus8 equal to 0 only.

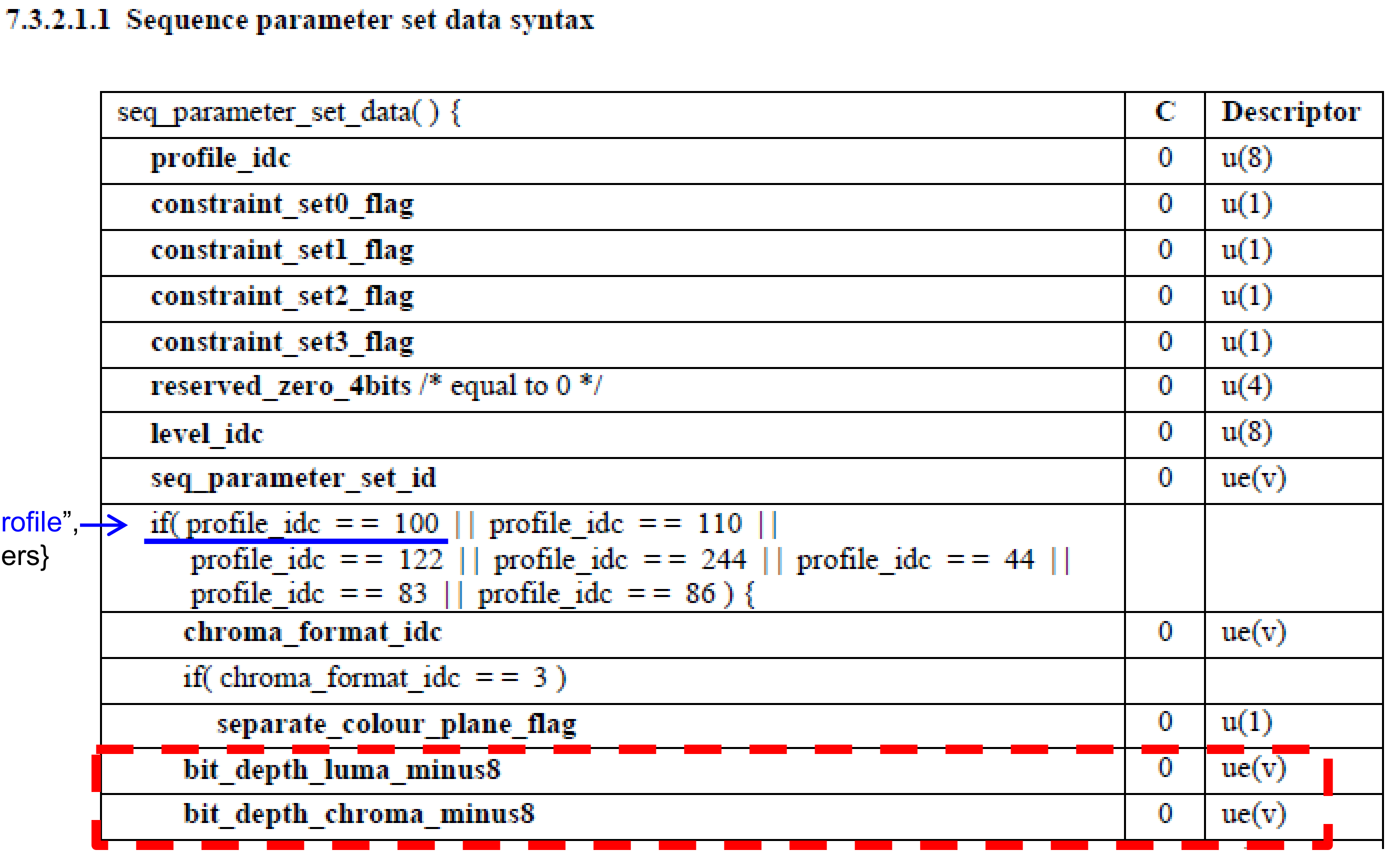
Preliminary Claim Chart Showing Infringement of Claim 1 of the U.S. Patent No. 6,744,818 by Lenovo

Claim 1

a visual perception estimator adapted to estimate a perception threshold for a pixel of a current frame of a videostream;

https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-H.264-200711-S!!PDF-E&type=items pg. 40

The “High Profile” includes the use of bit_depth_luma and bit_depth_chroma, which are used for the visual perception estimator:

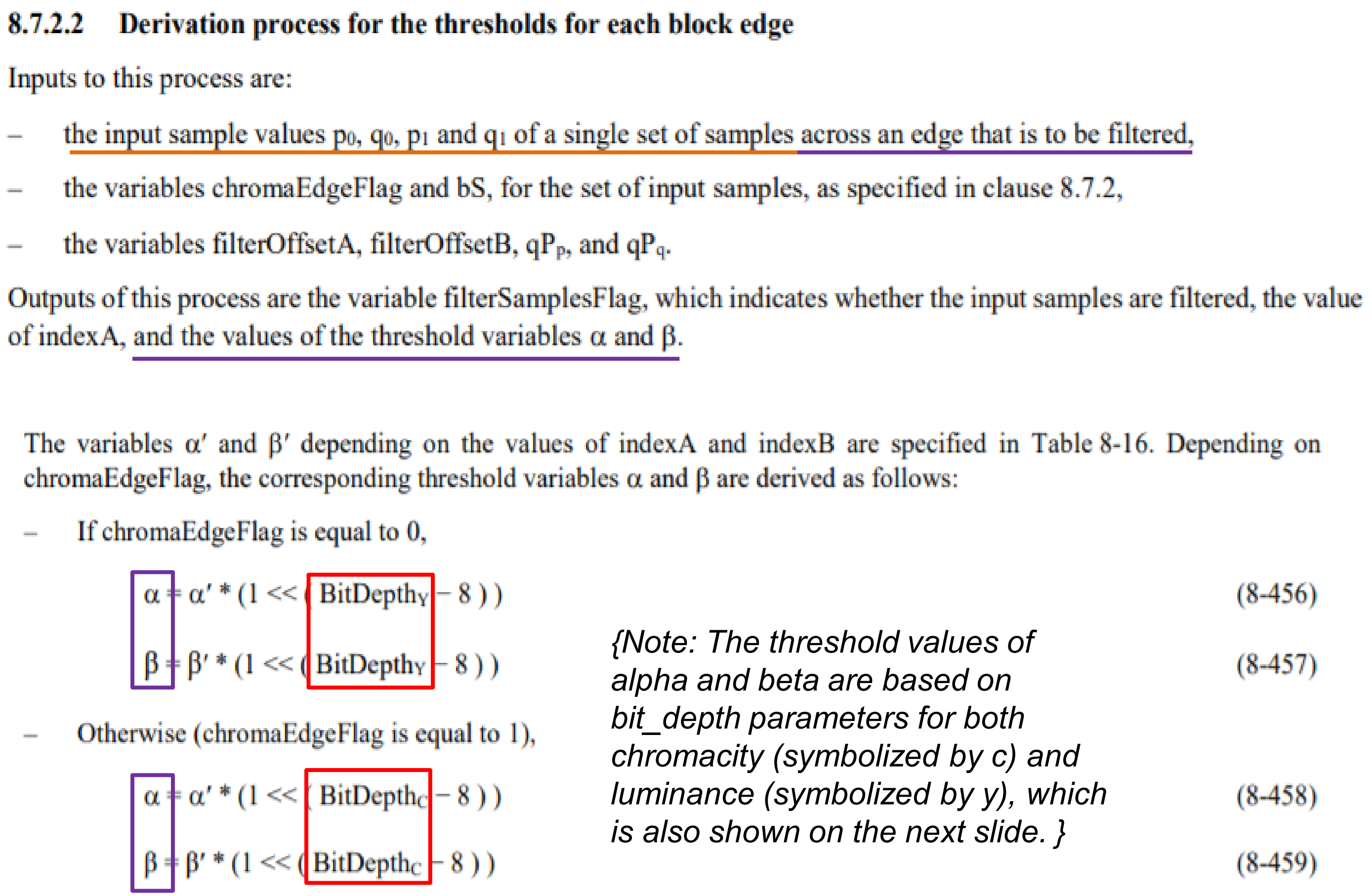


7.3.2.1.1 Sequence parameter set data syntax

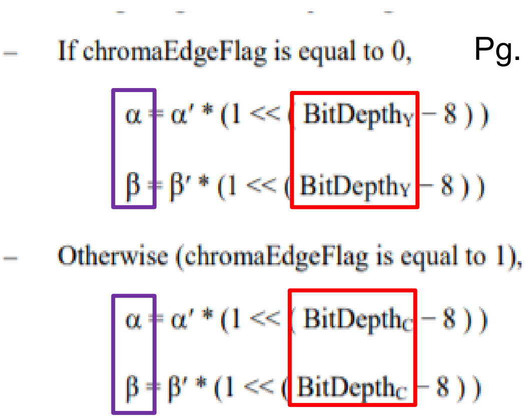
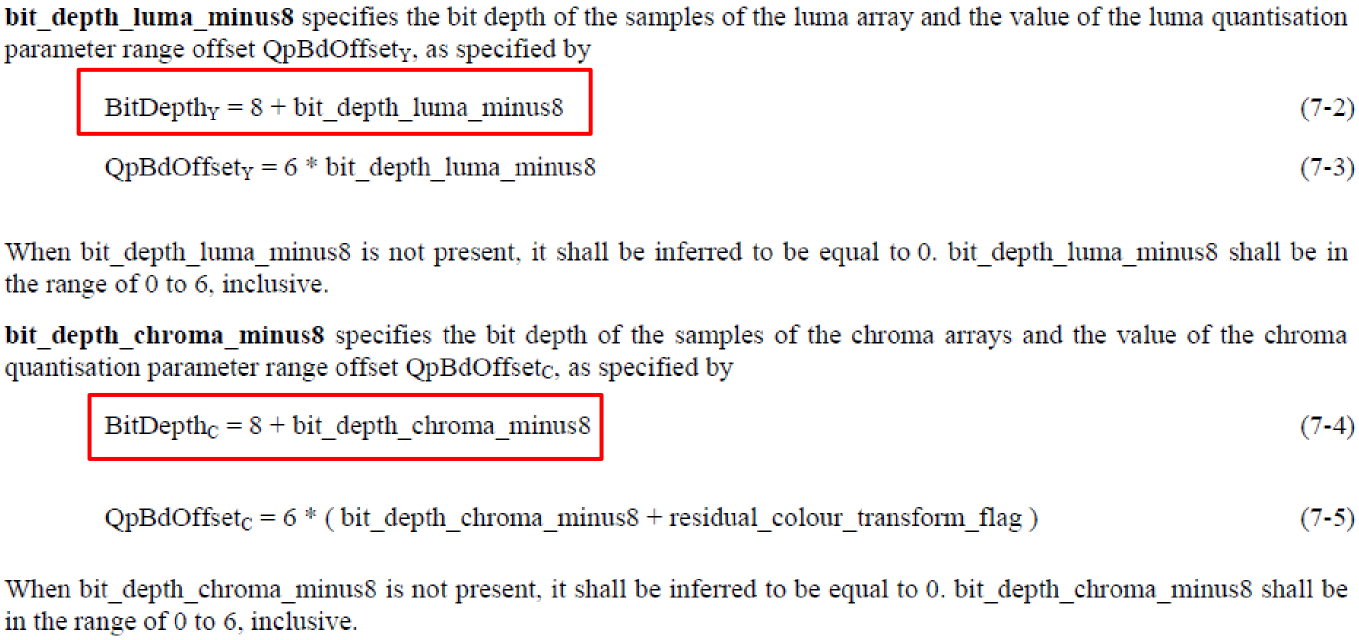
seq_parameter_set_data( ) {	C	Descriptor
profile_idc	0	u(8)
constraint_set0_flag	0	u(1)
constraint_set1_flag	0	u(1)
constraint_set2_flag	0	u(1)
constraint_set3_flag	0	u(1)
reserved_zero_4bits /* equal to 0 */	0	u(4)
level_idc	0	u(8)
seq_parameter_set_id	0	ue(v)
if(profile_idc = = 100 || profile_idc = = 110 || profile_idc = = 122 || profile_idc = = 244 || profile_idc = = 44 || profile_idc = = 83 || profile_idc = = 86) {		
chroma_format_idc	0	ue(v)
if(chroma_format_idc = = 3)		
separate_colour_plane_flag	0	u(1)
bit_depth_luma_minus8	0	ue(v)
bit_depth_chroma_minus8	0	ue(v)

{Note: “High Profile”, and others}

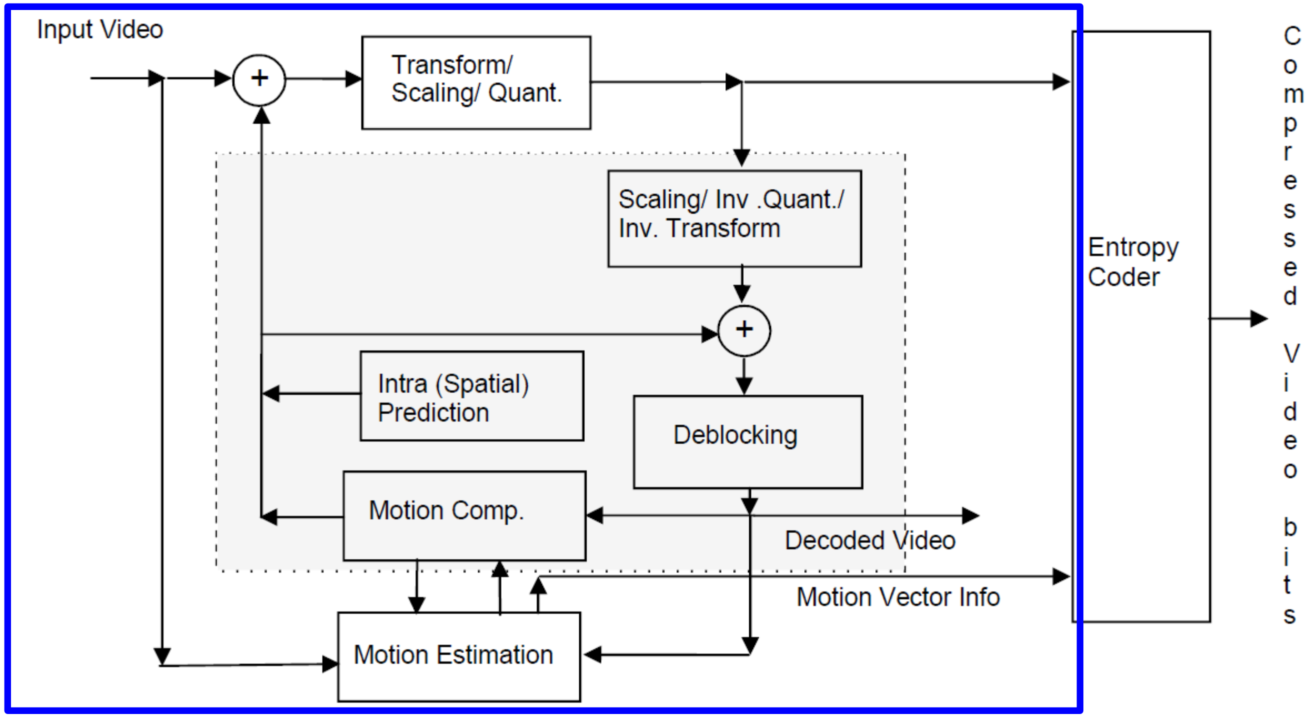
Preliminary Claim Chart Showing Infringement of Claim 1 of the U.S. Patent No. 6,744,818 by Lenovo

Claim 1	
a visual perception estimator adapted to estimate a perception threshold for a pixel of a current frame of a videostream;	https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-H.264-200711-S!!PDF-E&type=items pg. 201 - 202  **8.7.2.2 Derivation process for the thresholds for each block edge** Inputs to this process are: – the input sample values p_0, q_0, p_1 and q_1 of a single set of samples across an edge that is to be filtered, – the variables chromaEdgeFlag and bS, for the set of input samples, as specified in clause 8.7.2, – the variables filterOffsetA, filterOffsetB, qP_p, and qP_q. Outputs of this process are the variable filterSamplesFlag, which indicates whether the input samples are filtered, the value of indexA, and the values of the threshold variables α and β. The variables α′ and β′ depending on the values of indexA and indexB are specified in Table 8-16. Depending on chromaEdgeFlag, the corresponding threshold variables α and β are derived as follows: – If chromaEdgeFlag is equal to 0, $\alpha = \alpha' * (1 << (\mathrm{BitDepth}_Y - 8))$ (8-456) $\beta = \beta' * (1 << (\mathrm{BitDepth}_Y - 8))$ (8-457) – Otherwise (chromaEdgeFlag is equal to 1), $\alpha = \alpha' * (1 << (\mathrm{BitDepth}_C - 8))$ (8-458) $\beta = \beta' * (1 << (\mathrm{BitDepth}_C - 8))$ (8-459) *{Note: The threshold values of alpha and beta are based on bit_depth parameters for both chromacity (symbolized by c) and luminance (symbolized by y), which is also shown on the next slide. }*

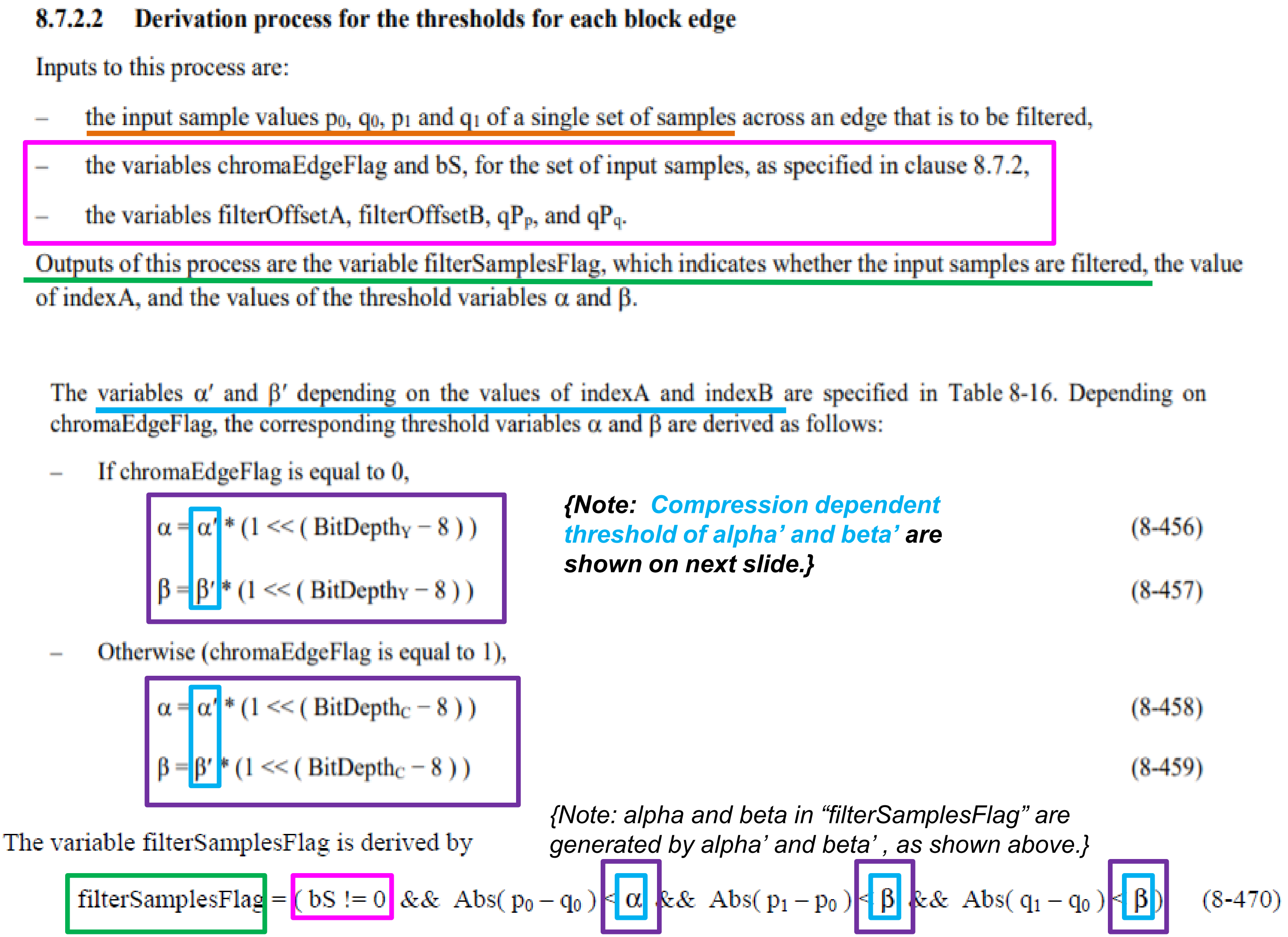
Preliminary Claim Chart Showing Infringement of Claim 1 of the U.S. Patent No. 6,744,818 by Lenovo

Claim 1	
a visual perception estimator adapted to estimate a perception threshold for a pixel of a current frame of a videostream;	 – If chromaEdgeFlag is equal to 0, $\alpha = \alpha' * (1 << (\mathrm{BitDepth}_Y - 8))$ $\beta = \beta' * (1 << (\mathrm{BitDepth}_Y - 8))$ – Otherwise (chromaEdgeFlag is equal to 1), $\alpha = \alpha' * (1 << (\mathrm{BitDepth}_C - 8))$ $\beta = \beta' * (1 << (\mathrm{BitDepth}_C - 8))$ Pg. 202 https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-H.264-200711-S!!PDF-E&type=items Pg. 67-68  **bit_depth_luma_minus8** specifies the bit depth of the samples of the luma array and the value of the luma quantisation parameter range offset $\mathrm{QpBdOffset}_Y$, as specified by $\boxed{\mathrm{BitDepth}_Y = 8 + \mathrm{bit\_depth\_luma\_minus8}}$ (7-2) $\mathrm{QpBdOffset}_Y = 6 * \mathrm{bit\_depth\_luma\_minus8}$ (7-3) When bit_depth_luma_minus8 is not present, it shall be inferred to be equal to 0. bit_depth_luma_minus8 shall be in the range of 0 to 6, inclusive. **bit_depth_chroma_minus8** specifies the bit depth of the samples of the chroma arrays and the value of the chroma quantisation parameter range offset $\mathrm{QpBdOffset}_C$, as specified by $\boxed{\mathrm{BitDepth}_C = 8 + \mathrm{bit\_depth\_chroma\_minus8}}$ (7-4) $\mathrm{QpBdOffset}_C = 6 * (\mathrm{bit\_depth\_chroma\_minus8} + \mathrm{residual\_colour\_transform\_flag})$ (7-5) When bit_depth_chroma_minus8 is not present, it shall be inferred to be equal to 0. bit_depth_chroma_minus8 shall be in the range of 0 to 6, inclusive.

Preliminary Claim Chart Showing Infringement of Claim 1 of the U.S. Patent No. 6,744,818 by Lenovo

Claim 1	
an encoder adapted to encode said current frame;	http://www.fastvdo.com/spie04/spie04-h264OverviewPaper.pdf  Input Video + Transform/ Scaling/ Quant. Scaling/ Inv .Quant./ Inv. Transform + Intra (Spatial) Prediction Deblocking Motion Comp. Decoded Video Motion Vector Info Motion Estimation Entropy Coder Compressed Video bits **Fig. 1: High-level encoder architecture**

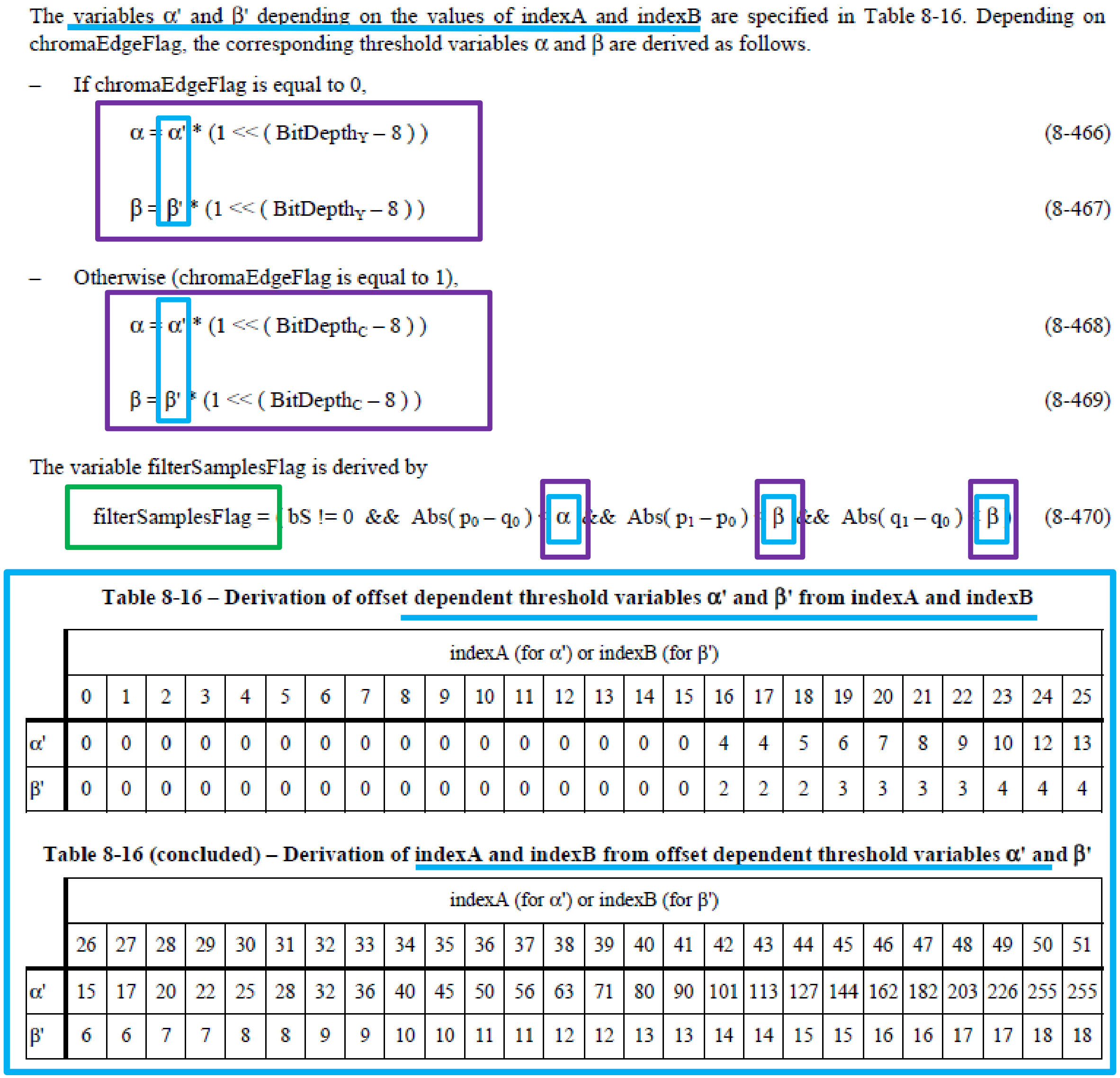
Preliminary Claim Chart Showing Infringement of Claim 1 of the U.S. Patent No. 6,744,818 by Lenovo

Claim 1	
a compression dependent threshold estimator adapted to estimate a compression dependent threshold for said pixel at least from said perception threshold and information from said encoder; and	https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-H.264-200711-S!!PDF-E&type=items pg. 201 - 202  **8.7.2.2 Derivation process for the thresholds for each block edge** Inputs to this process are: – the input sample values p_0, q_0, p_1 and q_1 of a single set of samples across an edge that is to be filtered, – the variables chromaEdgeFlag and bS, for the set of input samples, as specified in clause 8.7.2, – the variables filterOffsetA, filterOffsetB, qP_p, and qP_q. Outputs of this process are the variable filterSamplesFlag, which indicates whether the input samples are filtered, the value of indexA, and the values of the threshold variables α and β. The variables α′ and β′ depending on the values of indexA and indexB are specified in Table 8-16. Depending on chromaEdgeFlag, the corresponding threshold variables α and β are derived as follows: – If chromaEdgeFlag is equal to 0, $\alpha = \boxed{\alpha'} * (1 << (\mathrm{BitDepth}_Y - 8))$ (8-456) $\beta = \boxed{\beta'} * (1 << (\mathrm{BitDepth}_Y - 8))$ (8-457) *{Note: **Compression dependent threshold of alpha' and beta'** are shown on next slide.}* – Otherwise (chromaEdgeFlag is equal to 1), $\alpha = \boxed{\alpha'} * (1 << (\mathrm{BitDepth}_C - 8))$ (8-458) $\beta = \boxed{\beta'} * (1 << (\mathrm{BitDepth}_C - 8))$ (8-459) *{Note: alpha and beta in "filterSamplesFlag" are generated by alpha' and beta', as shown above.}* The variable filterSamplesFlag is derived by $\boxed{\mathrm{filterSamplesFlag}} = (\boxed{\mathrm{bS} \mathrel{!=} 0} \mathrel{\&\&} \mathrm{Abs}(p_0 - q_0) < \boxed{\alpha} \mathrel{\&\&} \mathrm{Abs}(p_1 - p_0) < \boxed{\beta} \mathrel{\&\&} \mathrm{Abs}(q_1 - q_0) < \boxed{\beta})$ (8-470)

Preliminary Claim Chart Showing Infringement of Claim 1 of the U.S. Patent No. 6,744,818 by Lenovo

Claim 1

a compression dependent threshold estimator adapted to estimate a compression dependent threshold for said pixel at least from said perception threshold and information from said encoder; and



The variables α' and β' depending on the values of indexA and indexB are specified in Table 8-16. Depending on chromaEdgeFlag, the corresponding threshold variables α and β are derived as follows.

– If chromaEdgeFlag is equal to 0,

$$\alpha = \alpha' * (1 << (\mathrm{BitDepth}_Y - 8)) \qquad (8\text{-}466)$$

$$\beta = \beta' * (1 << (\mathrm{BitDepth}_Y - 8)) \qquad (8\text{-}467)$$

– Otherwise (chromaEdgeFlag is equal to 1),

$$\alpha = \alpha' * (1 << (\mathrm{BitDepth}_C - 8)) \qquad (8\text{-}468)$$

$$\beta = \beta' * (1 << (\mathrm{BitDepth}_C - 8)) \qquad (8\text{-}469)$$

The variable filterSamplesFlag is derived by

$$\mathrm{filterSamplesFlag} = (\mathrm{bS} \mathrel{!=} 0 \ \&\& \ \mathrm{Abs}(p_0 - q_0) < \alpha \ \&\& \ \mathrm{Abs}(p_1 - p_0) < \beta \ \&\& \ \mathrm{Abs}(q_1 - q_0) < \beta) \qquad (8\text{-}470)$$

Table 8-16 – Derivation of offset dependent threshold variables α' and β' from indexA and indexB

	indexA (for α') or indexB (for β')																									
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
α'	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	5	6	7	8	9	10	12	13
β'	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	3	3	3	3	4	4	4

Table 8-16 (concluded) – Derivation of indexA and indexB from offset dependent threshold variables α' and β'

	indexA (for α') or indexB (for β')																									
	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
α'	15	17	20	22	25	28	32	36	40	45	50	56	63	71	80	90	101	113	127	144	162	182	203	226	255	255
β'	6	6	7	7	8	8	9	9	10	10	11	11	12	12	13	13	14	14	15	15	16	16	17	17	18	18

https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-H.264-200711-S!!PDF-E&type=items

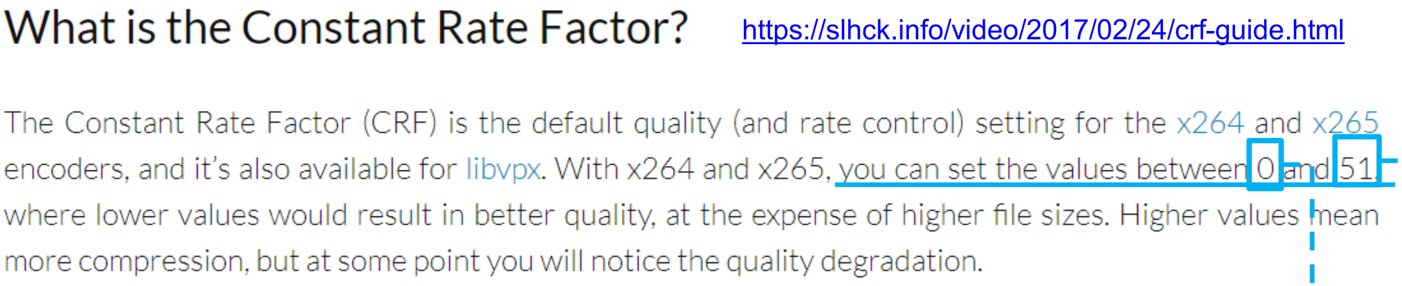
Pg. 202

{Note: The values of IndexA and IndexB from "0" to "51" are for the "QP" value or "quantization parameter", which specifies the level of compression. See next slide}

Preliminary Claim Chart Showing Infringement of Claim 1 of the U.S. Patent No. 6,744,818 by Lenovo

Claim 1

a compression dependent threshold estimator adapted to estimate a compression dependent threshold for said pixel at least from said perception threshold and information from said encoder; and

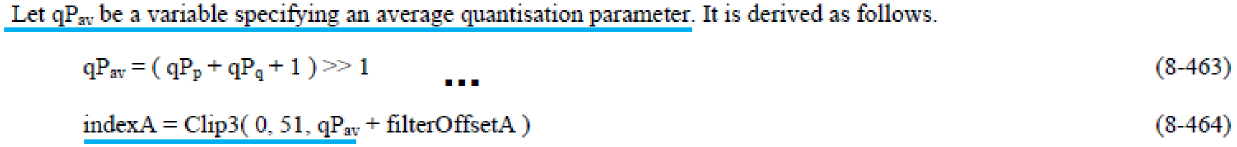


What is the Constant Rate Factor?

https://slhck.info/video/2017/02/24/crf-guide.html

The Constant Rate Factor (CRF) is the default quality (and rate control) setting for the x264 and x265 encoders, and it's also available for libvpx. With x264 and x265, you can set the values between 0 and 51, where lower values would result in better quality, at the expense of higher file sizes. Higher values mean more compression, but at some point you will notice the quality degradation.

https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-H.264-200711-S!!PDF-E&type=items pg. 201



Let qP_{av} be a variable specifying an average quantisation parameter. It is derived as follows.

$$qP_{av} = (qP_p + qP_q + 1) >> 1 \quad (8\text{-}463)$$

...

$$indexA = Clip3(0, 51, qP_{av} + filterOffsetA) \quad (8\text{-}464)$$

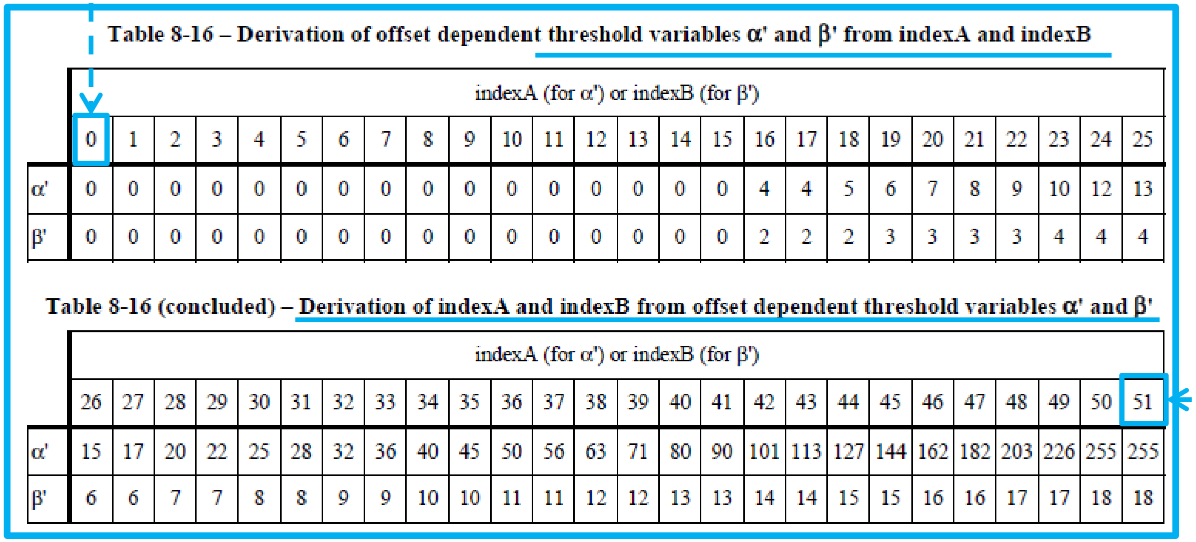


Table 8-16 – Derivation of offset dependent threshold variables α' and β' from indexA and indexB

	indexA (for α') or indexB (for β')																									
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
α'	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	5	6	7	8	9	10	12	13
β'	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	3	3	3	3	4	4	4

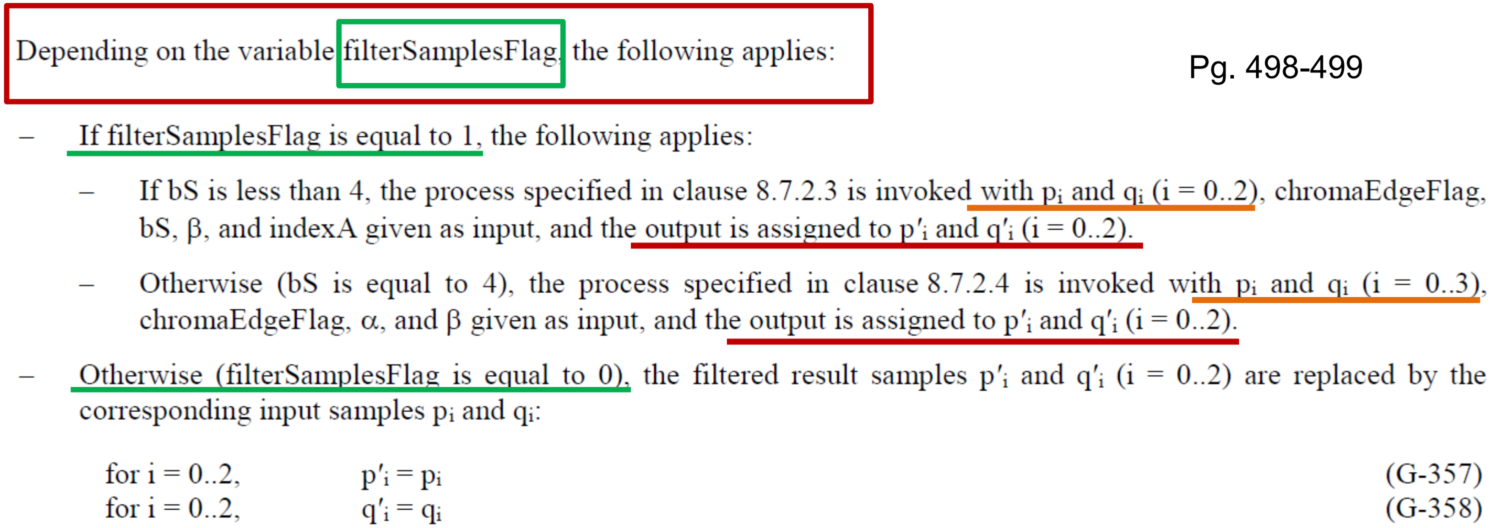
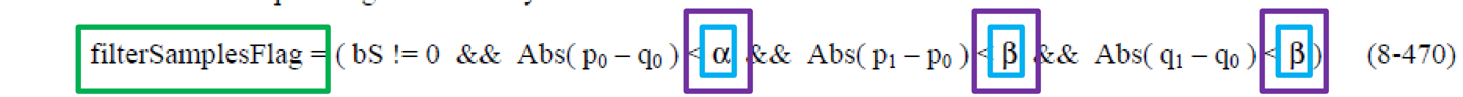
Table 8-16 (concluded) – Derivation of indexA and indexB from offset dependent threshold variables α' and β'

	indexA (for α') or indexB (for β')																									
	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
α'	15	17	20	22	25	28	32	36	40	45	50	56	63	71	80	90	101	113	127	144	162	182	203	226	255	255
β'	6	6	7	7	8	8	9	9	10	10	11	11	12	12	13	13	14	14	15	15	16	16	17	17	18	18

Pg. 202

{Note: The values "0" to "51" specify the level of compression. }

Preliminary Claim Chart Showing Infringement of Claim 1 of the U.S. Patent No. 6,744,818 by Lenovo

Claim 1	
a filter unit adapted to filter said pixel at least according to said compression dependent threshold.	https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-H.264-200711-S!!PDF-E&type=items ***{Note: “filterSampleFlag” is compression dependent threshold estimator for compression dependent threshold of alpha’ and beta’, as shown on slide 11. }*** **G.8.7.4.2 SVC filtering process for a set of samples across a horizontal or vertical block edge …**  Depending on the variable filterSamplesFlag, the following applies: Pg. 498-499 – If filterSamplesFlag is equal to 1, the following applies: – If bS is less than 4, the process specified in clause 8.7.2.3 is invoked with p_i and q_i (i = 0..2), chromaEdgeFlag, bS, β, and indexA given as input, and the output is assigned to p'_i and q'_i (i = 0..2). – Otherwise (bS is equal to 4), the process specified in clause 8.7.2.4 is invoked with p_i and q_i (i = 0..3), chromaEdgeFlag, α, and β given as input, and the output is assigned to p'_i and q'_i (i = 0..2). – Otherwise (filterSamplesFlag is equal to 0), the filtered result samples p'_i and q'_i (i = 0..2) are replaced by the corresponding input samples p_i and q_i: for i = 0..2, $p'_i = p_i$ (G-357) for i = 0..2, $q'_i = q_i$ (G-358) Pg. 202 The variable filterSamplesFlag is derived by  filterSamplesFlag = (bS != 0 && Abs($p_0 - q_0$) < α && Abs($p_1 - p_0$) < β && Abs($q_1 - q_0$) < β) (8-470)